

# Tactical Security

## Improving AppSec Coverage with Fewer Resources

Rich Newman, CISSP, Sales Engineer, Synopsys Software Integrity Group

May 2023



# Rich Newman



13 years in software development

8 years field engineering,  
Wind River Operating Systems and Tools

6 years field engineering,  
Wind River Test Management

11 years (minus 12 days) field engineering,  
Coverity and Synopsys

[rnewman@synopsys.com](mailto:rnewman@synopsys.com)

949.466.5283

# Today's Agenda

1. Software Development Process
2. SAST
3. SCA
4. DAST (IAST/Fuzz)
5. ASOC
6. Help!

SAST - Static Application Security Testing

SCA - Software Composition Analysis

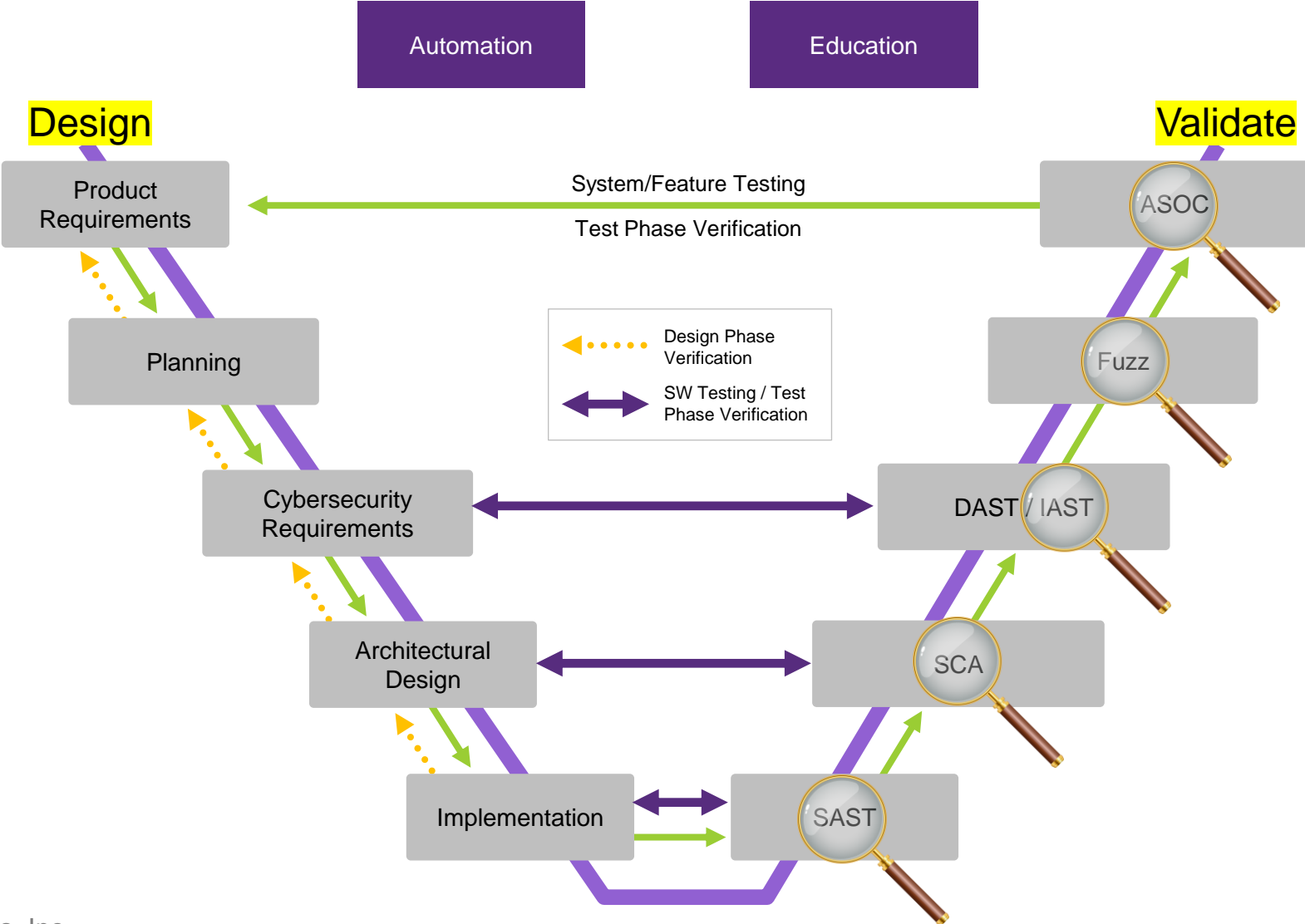
DAST - Dynamic Application Security Testing

IAST - Interactive Application Security Testing

Fuzz - Malformed protocol testing

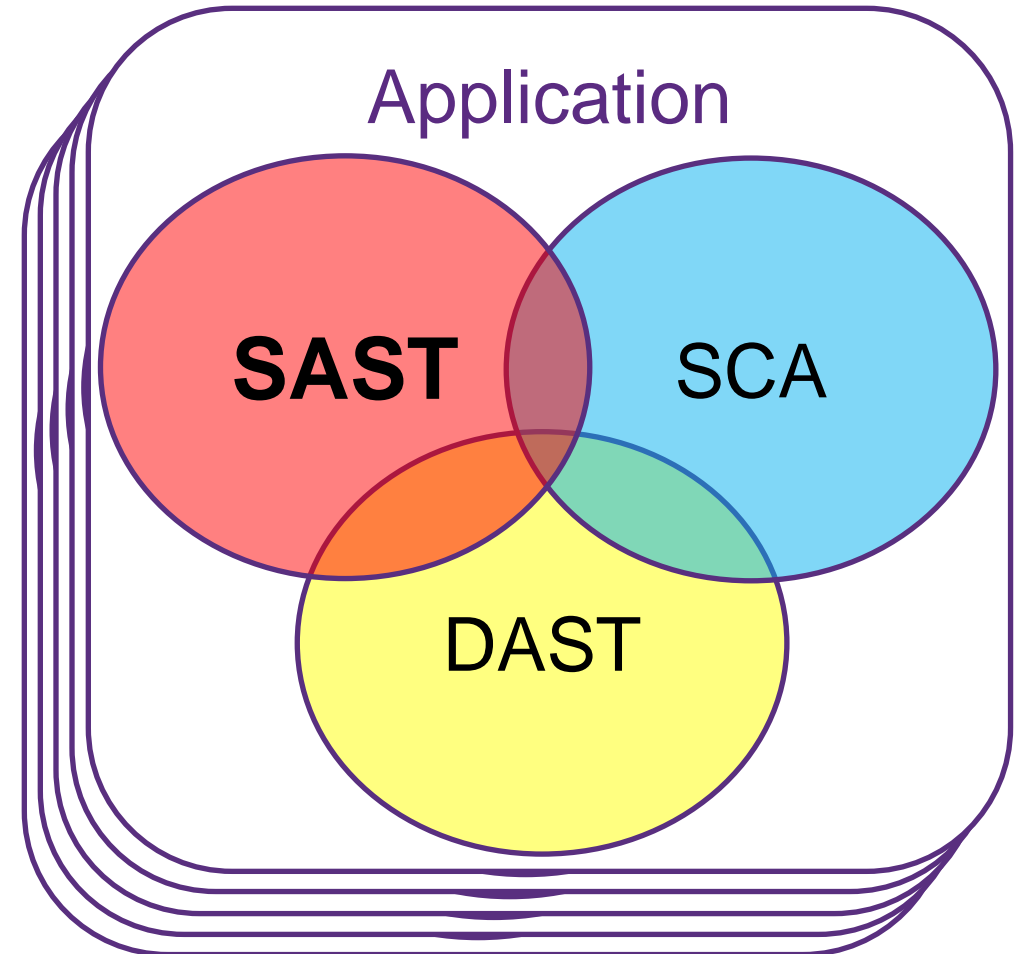
ASOC - Application Security Orchestration and Correlation

# Software Development Process



# Today's Agenda

1. Software Development Process
2. SAST
3. SCA
4. DAST (IAST/Fuzz)
5. ASOC
6. Help!



# Simple Web Application

YourWebProject2

## Register

Display Name

Email

Password

Verify

REGISTER

CANCEL

YourWebProject2

## Login

Email

Password

LOGIN

REGISTER

YourWebProject2

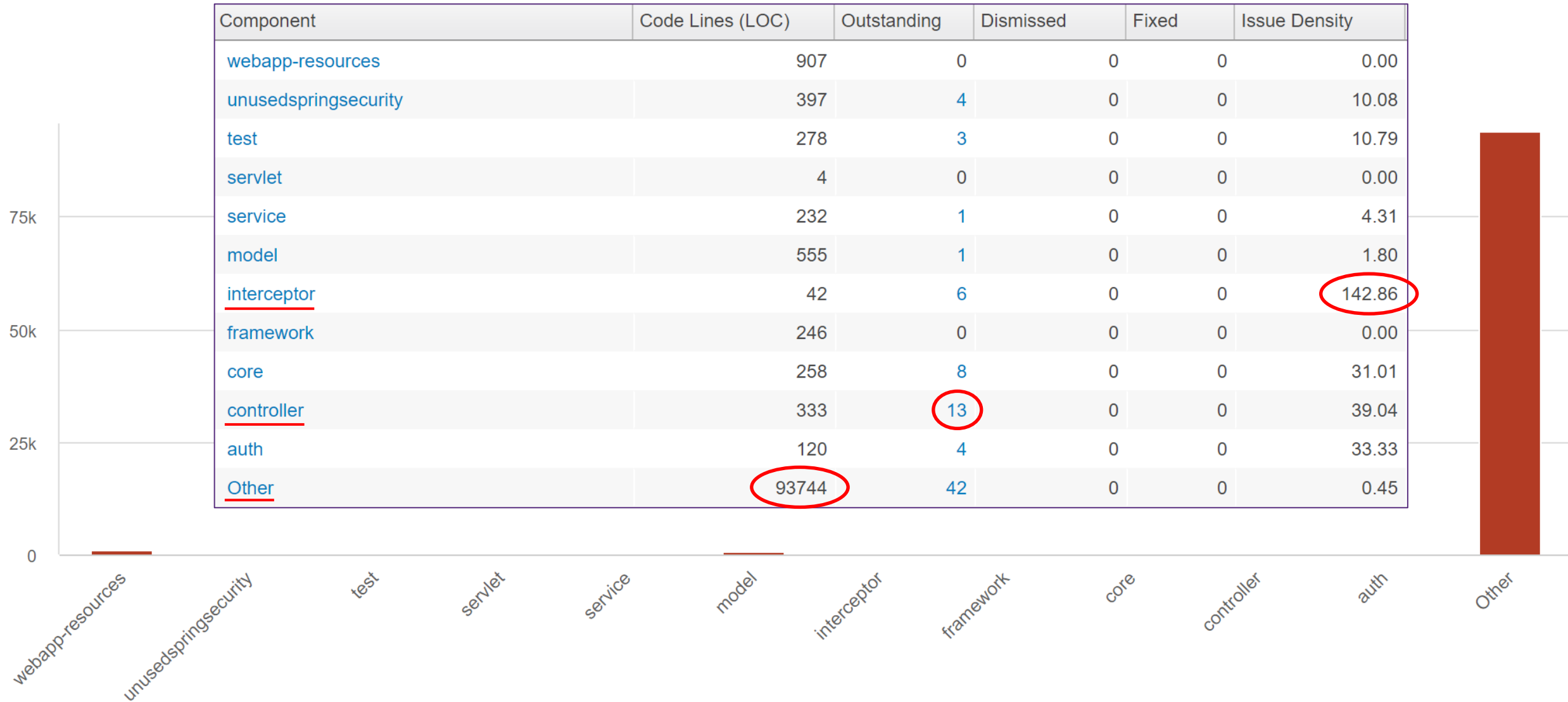
YOUR AWESOME MOTTO GOES HERE...!

SIGN IN

## Dashboard

© 2015, Kaleidosoft Labs

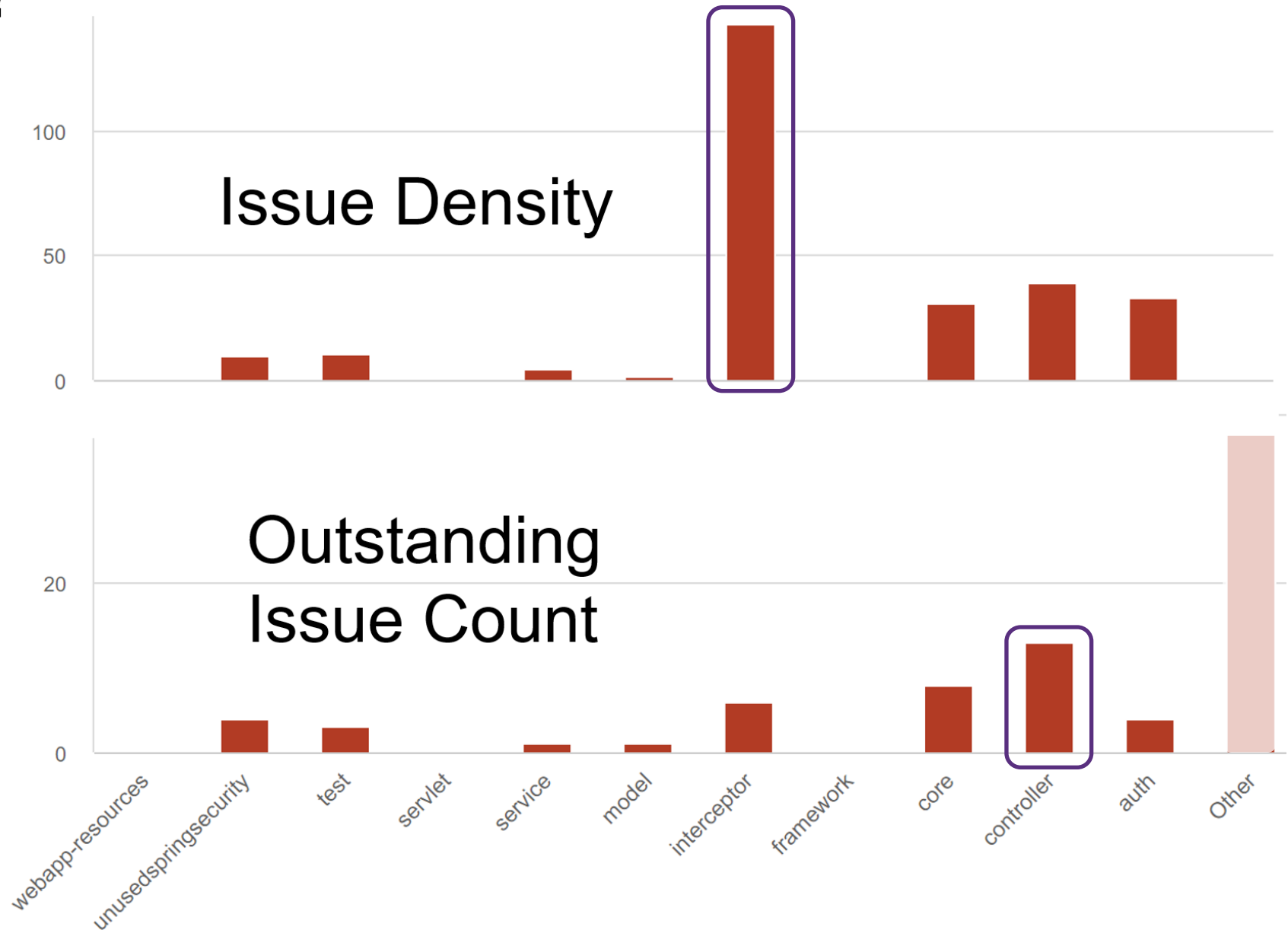
# What are the Components?



# What are the risks?

Many issues in a small area of the application may indicate a design concern (fever)

Too many issues reveals high technical debt (general malise)





# Issue Density

Writing invalidated user input to log files can allow an attacker to forge log entries or inject malicious content into the logs

- Injection of misleading events
- Injection of XSS attacks, hoping that the malicious log event is viewed in a vulnerable web application
- Injection of commands that parsers (like PHP parsers) could execute

The screenshot shows a web application security tool interface. At the top, there's a header with 'Webapp' and 'Issues: By Snapshot | All'. Below this is a table of issues:

Component	# Items	CID	Category	Type	Standard: OWASP Web Top Ten 2021	CWE
WebApp.test	3	139639	Audit impact security	Log injection	A3	117
WebApp.interceptor	6	139641	Audit impact security	Log injection	A3	117
WebApp.auth	4	139642	Audit impact security	Log injection	A3	117
WebApp.Other	42	139643	Audit impact security	Log injection	A3	117
WebApp.service	1	139645	Audit impact security	Log injection	A3	117

Below the table, there's a code editor showing the source code for 'WebAppMetricsInterceptor.java'. The code includes a `preHandle` method with several security warnings:

- CID 139641: Log injection (LOG\_INJECTION) [select issue]
- CID 139645: Log injection (LOG\_INJECTION) [select issue]
- 1. **argument\_audit**: handler may assume any value when called by an unknown or untrusted caller.
- CID 139652: Log injection (LOG\_INJECTION) [select issue]
- 2. **identity**: Calling toString. This call assigns handler to <return value>. Now <return value> is tainted. (The virtual call resolves to java.lang.Object.toString().)
- 3. **concat**: Creating a tainted string using handler.toString().
- This is a security audit finding.
- CID 139643 (#1 of 1): Log injection (LOG\_INJECTION)
- 4. **sink**: Calling info. This call uses a tainted string for sensitive computation. (The virtual call resolves to org.slf4j.Logger.info(java.lang.String).)
- Log injection vulnerabilities can be addressed by validating that the user-controllable input conforms to expectations.

```
18 private long startTime = 0L;
19
20 @Override
21 public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler) throws Exception {
22     LOG.info("processing: " + request.getRequestURI() + " Handler: " + handler.toString());
23     startTime = System.currentTimeMillis();
24     return super.preHandle(request, response, handler);
25 }
26
27 @Override
28 public void postHandle(HttpServletRequest request, HttpServletResponse response, Object handler, ModelAndView modelAndView) throws Exception {
29     super.postHandle(request, response, handler, modelAndView);
30 }
31
32 LOG.info(String.format("responseTime: %d ms", responseTime));
33 }
34 }
```

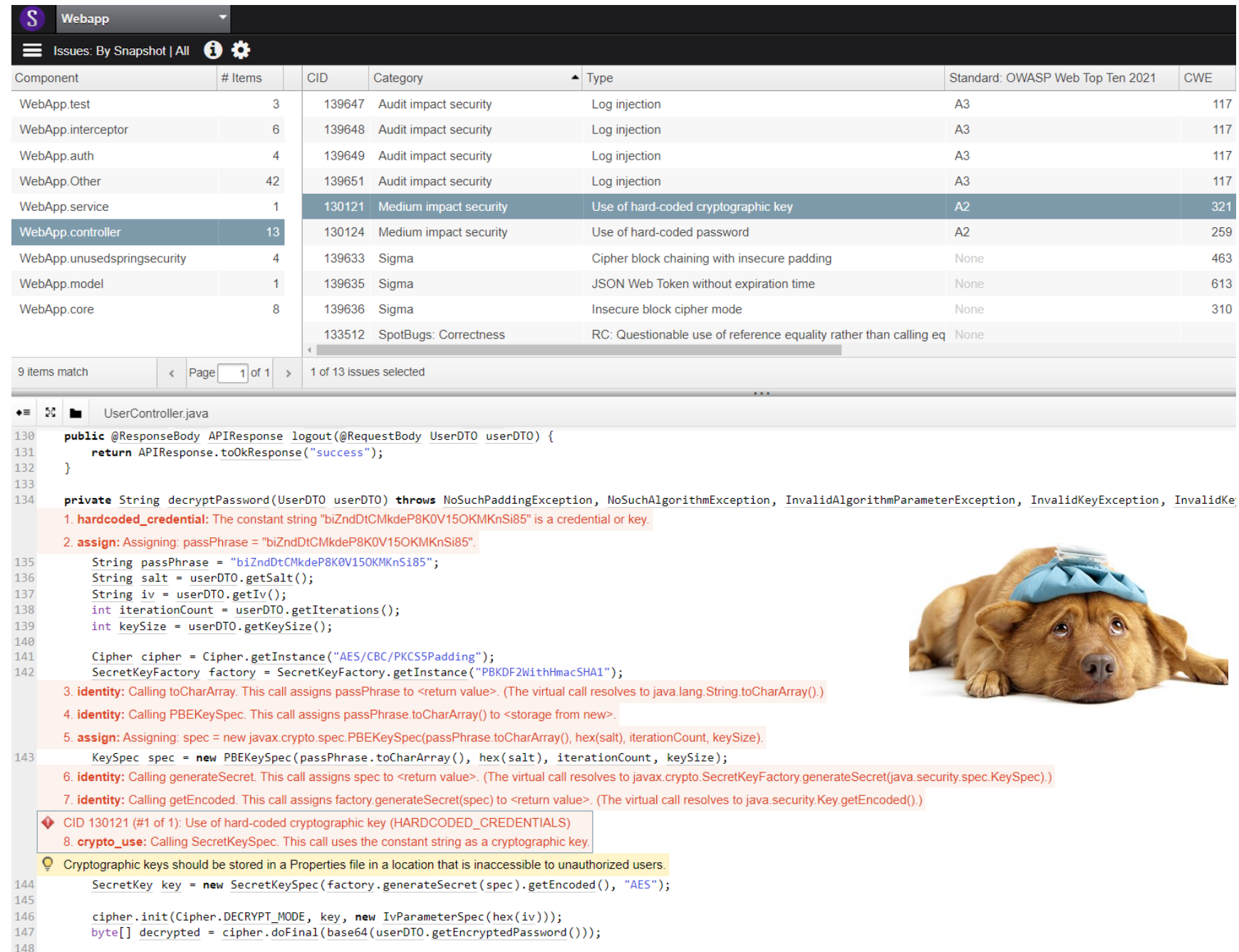


# Technical Debt

Any code that decreases agility as the project matures

Agility decreases as additional time is needed to investigate, resolve and test delayed issue resolution

Another task on the things to do list: Develop an inaccessible properties file



The screenshot displays a web application security scanner interface. At the top, there's a browser tab labeled 'Webapp' and a navigation bar with 'Issues: By Snapshot | All' and icons for help and settings. Below this is a table listing various security issues. The table has columns for Component, # Items, CID, Category, Type, Standard: OWASP Web Top Ten 2021, and CWE. The 'WebApp.controller' component is highlighted, showing 13 items. One issue, CID 130121, is selected, which is a 'Medium impact security' issue related to the 'Use of hard-coded cryptographic key' (A2 standard, CWE 321).

Component	# Items	CID	Category	Type	Standard: OWASP Web Top Ten 2021	CWE
WebApp.test	3	139647	Audit impact security	Log injection	A3	117
WebApp.interceptor	6	139648	Audit impact security	Log injection	A3	117
WebApp.auth	4	139649	Audit impact security	Log injection	A3	117
WebApp.Other	42	139651	Audit impact security	Log injection	A3	117
WebApp.service	1	130121	Medium impact security	Use of hard-coded cryptographic key	A2	321
WebApp.controller	13	130124	Medium impact security	Use of hard-coded password	A2	259
WebApp.unusedspringsecurity	4	139633	Sigma	Cipher block chaining with insecure padding	None	463
WebApp.model	1	139635	Sigma	JSON Web Token without expiration time	None	613
WebApp.core	8	139636	Sigma	Insecure block cipher mode	None	310
		133512	SpotBugs: Correctness	RC: Questionable use of reference equality rather than calling eq	None	

Below the table, there's a navigation bar showing '9 items match' and 'Page 1 of 1'. The main content area shows a code editor for 'UserController.java'. The code is annotated with several security issues:

```
130 public @ResponseBody ApiResponse logout(@RequestBody UserDTO userDTO) {
131     return ApiResponse.toOkResponse("success");
132 }
133
134 private String decryptPassword(UserDTO userDTO) throws NoSuchPaddingException, NoSuchAlgorithmException, InvalidAlgorithmParameterException, InvalidKeyException, InvalidKeyException {
135     String passPhrase = "biZndDtCMkdeP8K0V15OKMKnSi85";
136     String salt = userDTO.getSalt();
137     String iv = userDTO.getIv();
138     int iterationCount = userDTO.getIterations();
139     int keySize = userDTO.getKeySize();
140
141     Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
142     SecretKeyFactory factory = SecretKeyFactory.getInstance("PBKDF2WithHmacSHA1");
143     KeySpec spec = new PBEKeySpec(passPhrase.toCharArray(), hex(salt), iterationCount, keySize);
144     SecretKey key = new SecretKeySpec(factory.generateSecret(spec).getEncoded(), "AES");
145
146     cipher.init(Cipher.DECRYPT_MODE, key, new IvParameterSpec(hex(iv)));
147     byte[] decrypted = cipher.doFinal(base64(userDTO.getEncryptedPassword()));
148 }
```

The annotations include:

- 1. **hardcoded\_credential**: The constant string "biZndDtCMkdeP8K0V15OKMKnSi85" is a credential or key.
- 2. **assign**: Assigning: passPhrase = "biZndDtCMkdeP8K0V15OKMKnSi85".
- 3. **identity**: Calling toCharArray. This call assigns passPhrase to <return value>. (The virtual call resolves to java.lang.String.toCharArray().)
- 4. **identity**: Calling PBEKeySpec. This call assigns passPhrase.toCharArray() to <storage from new>.
- 5. **assign**: Assigning: spec = new javax.crypto.spec.PBEKeySpec(passPhrase.toCharArray(), hex(salt), iterationCount, keySize).
- 6. **identity**: Calling generateSecret. This call assigns spec to <return value>. (The virtual call resolves to javax.crypto.SecretKeyFactory.generateSecret(javax.crypto.spec.KeySpec).
- 7. **identity**: Calling getEncoded. This call assigns factory.generateSecret(spec) to <return value>. (The virtual call resolves to java.security.Key.getEncoded().)
- 8. **crypto\_use**: Calling SecretKeySpec. This call uses the constant string as a cryptographic key.

A warning icon indicates a cryptographic key should be stored in a Properties file in a location that is inaccessible to unauthorized users.



Issues: By Snapshot | All

Component	# Items	CID	Category	Type
WebApp.test	3	139649	Audit impact security	Log injection
WebApp.interceptor	6	139651	Audit impact security	Log injection
WebApp.auth	4	130121	Medium impact security	Use of hard-coded cryptographic key
WebApp.Other	42	130124	Medium impact security	Use of hard-coded password
WebApp.service	1	139633	Sigma	Cipher block chaining with insecure padding
WebApp.controller	13	139635	Sigma	JSON Web Token without expiration time
WebApp.unusedspringsecurity	4	139636	Sigma	Insecure block cipher mode
WebApp.model	1			

9 items match Page 1 of 1 1 of 13 issues selected

**139633 Cipher block chaining with insecure padding**

In yourwebproject2.controller.  
UserController::decryptPassword(yourwebproject2.controller.  
UserController, yourwebproject2.model.dto.UserDTO): Using a block cipher with `CBC` mode and `PKCS5Padding` may be vulnerable to padding oracle attacks. In many scenarios, an attacker can exploit this issue to recover the full plaintext with little effort. (CWE-463)

... Less

▼ Triage

Classification:

Severity:

Action:

Legacy:

# IaC – Infrastructure as Code

```

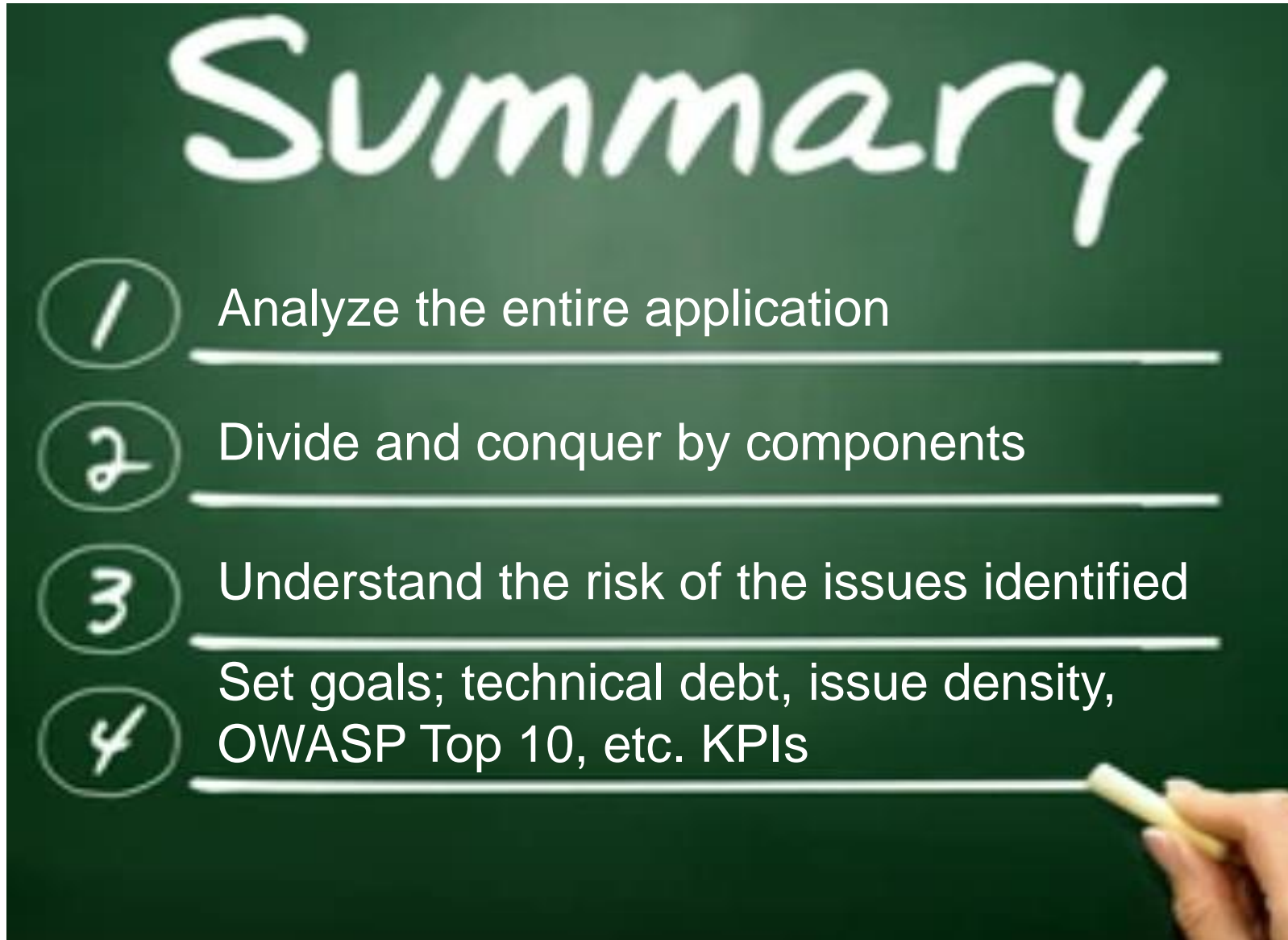
UserController.java
122
123 /**
124  * Logs out a user by deleting the session
125  *
126  * @param userDTO
127  * @return
128  */
129 @RequestMapping(value = "/logout", method = RequestMethod.DELETE)
130 public @ResponseBody APIResponse logout(@RequestBody UserDTO userDTO) {
131     return APIResponse.toOkResponse("success");
132 }
133
134 private String decryptPassword(UserDTO userDTO) throws NoSuchPaddingException, NoSuchAlgorithmException, InvalidAlgorithmParameterException, InvalidKeyException, InvalidKeySpecException {
135     String passPhrase = "biZndDtCMkdeP8K0V15OKMKnSi85";
136     String salt = userDTO.getSalt();
137     String iv = userDTO.getIv();
138     int iterationCount = userDTO.getIterations();
139     int keySize = userDTO.getKeySize();
140
141     Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");

```

◆ CID 139636: Insecure block cipher mode (SIGMA.insecure\_block\_cipher\_mode) [\[select issue\]](#)  
◆ CID 139633 (#1 of 1): Cipher block chaining with insecure padding (SIGMA.cbc\_insecure\_padding)  
 1. **Sigma main event:** Using a block cipher with `CBC` mode and `PKCS5Padding` may be vulnerable to padding oracle attacks. In many scenarios, an attacker can exploit this issue to recover the full plaintext with little effort.  
💡 Instead of using `CBC` mode with `PKCS5Padding`, consider using `CCM` mode with `NoPadding`. Alternatively, consider `GCM` mode with `NoPadding`.

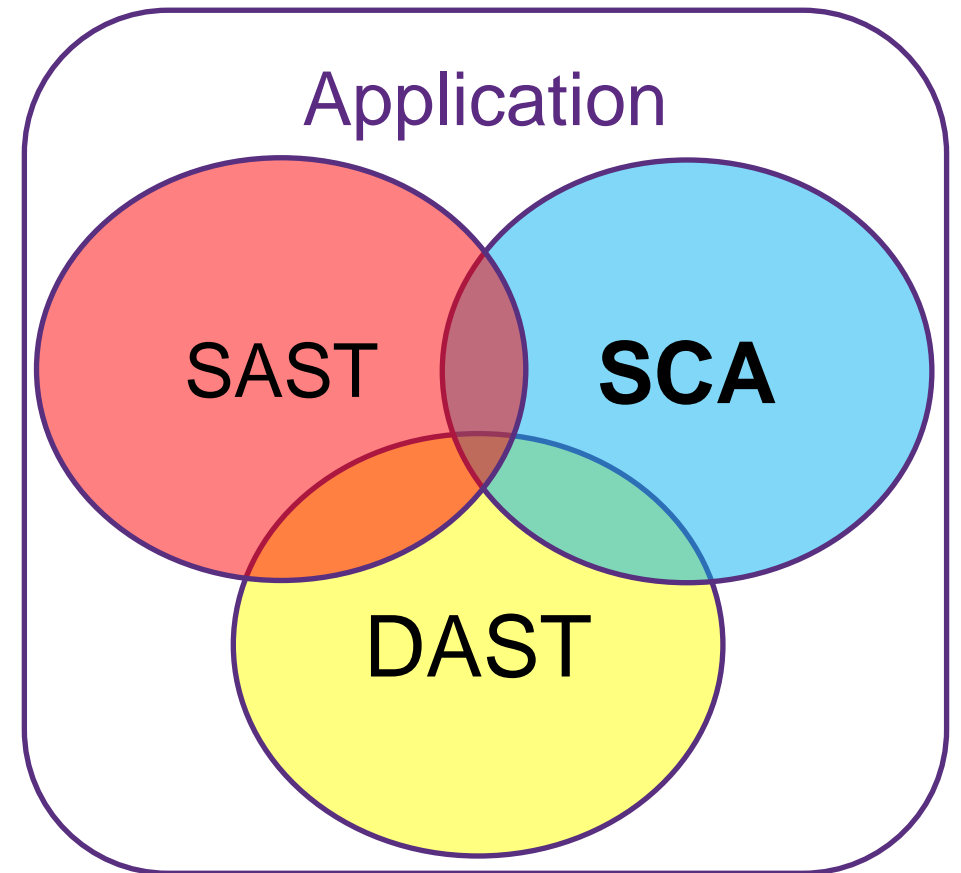


# Static Application Security Testing

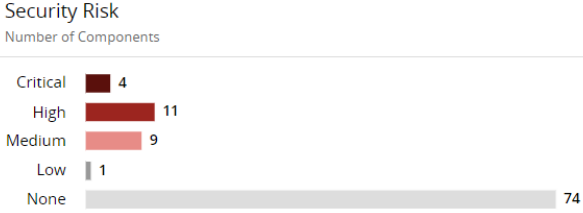


# Today's Agenda

1. Software Development Process
2. SAST
3. SCA
4. DAST (IAST/Fuzz)
5. ASOC
6. Help!



# Automating Risk Awareness



### Unmatched Components

0 Unmatched

▶ IaC  
9 Open

There are 25 components with security risk, how do we prioritize?

Buttons: Add, Bulk Actions, Compare to..., Print...  
Match ignore, Not ignored, Match Status, Confirmed, Ignore, Not Ignored, Filter Components...

Component	Source	Match Type	Usage	License	Security Risk	Operational Risk
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> jackson-databind 2.4.3	3 Matches	Direct Dependency, Exact Directory	Dynamically Linked	Apache-2.0	1 60 7	High
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> Spring Framework 4.1.6.RELEASE	30 Matches	Direct Dependency, Transitive Dependency, Exact Directory	Dynamically Linked	Apache-2.0	1 5 10 2	High
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> Spring Framework 4.0.9.RELEASE	3 Matches	Transitive Dependency, Exact Directory	Dynamically Linked	Apache-2.0	1 5 9 2	High
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> Logback 1.1.2	6 Matches	Direct Dependency, Exact Directory	Dynamically Linked	M LGPL-2.1 or 1 more...	1	High
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> Servlet, WebSocket Server 6.1.11	3 Matches	Direct Dependency, Exact Directory	Dynamically Linked	Apache-2.0	8 9 1	High
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> ...	2 Matches	Exact Directory	Dynamically Linked	Apache-2.0	4 5	High
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> ...	3 Matches	Exact Directory	Dynamically Linked	MIT	4	High
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> ...	3 Matches	Direct Dependency, Exact Directory	Dynamically Linked	M H2 License Version 1.0 or 1 more...	4	High
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> ...	9 Matches	Direct Dependency, Transitive Dependency, Exact Directory	Dynamically Linked	Apache-2.0	3 6	High
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> MySQL Connector/J 5.1.34	3 Matches	Direct Dependency, Exact Directory	Dynamically Linked	H GPL-2.0+	2 7 1	High
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> Data Mapper for Jackson 1.9.13	3 Matches	Direct Dependency, Exact Directory	Dynamically Linked	Apache-2.0	1 1	High
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> c3p0:JDBC DataSources/Resource Pools 0.9.2.1	3 Matches	Transitive Dependency, Exact Directory	Dynamically Linked	M LGPL-2.1+ or 1 more...	1 1	High
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> c3p0:JDBC DataSources/Resource Pools 0.9.1.2	3 Matches	Direct Dependency, Exact Directory	Dynamically Linked	M LGPL-2.1+	1 1	High

### Policy Violations

This component violates the following policies:

- Resolve
- Exploit
- Old and Vulnerable



# Jackson-databind Deserialization Remote Code Execution (RCE)

Triggered policy **Exploit**:

- ✓ Known exploit
- ✓ Zero-click RCE
- ✓ CVE  $\geq 9$



## Exploit

Severity: **Critical** Category: Security Scan Modes: Full, Rapid

### Description

CVE  $\geq 9$  with known exploit and RCE Zero-click

### Conditions

Exploit Available EQUALS True

Overall Score GREATER THAN OR EQUAL TO 9

Vulnerability Tags IN Zero-click Remote Code Execution

## Resolve

Severity: **Major** Category: Security Scan Modes: Full

### Description

CVE  $\geq 9$  and solution is available

### Conditions

Solution Available EQUALS True

Overall Score GREATER THAN OR EQUAL TO 9

## Old and Vulnerable

Severity: **Critical** Category: Security Scan Modes: Full

### Description

2015 or earlier and CVE  $> 9$

### Conditions

Component Release Date LESS THAN Jun 9, 2015

Overall Score GREATER THAN OR EQUAL TO 9

Identifier	Overall Score	Status	CWE	Exploit	Workaround	Solution
<b>BDSA</b> BDSA-2017-2725	<b>9.1</b> Critical	New	CWE-502	✓	-	✓

Zero-click RCE

### Description

Deserialization of untrusted user data in Jackson Databind could allow an attacker to perform Remote Code Execution via specially crafted JSON input.

This issue exists because of an incomplete fix for CVE-2017-7525 which the vendor tried to address through an incomplete blacklist.

[View BDSA record](#)

- ④ The Black Duck Security Advisory (BDSA) team mapped BDSA-2017-2725 to this component version, but it was not included in the National Vulnerability Database (NVD)'s associated record.

[Learn more about the benefits of BDSA](#)

**jackson-databind** 2.4.3  
maven:  
com.fasterxml.jackson.core:jackson-databind:2.4.3

Vulnerabilities **1** **60** **7**

# Research

CRITICAL 9.1  
BDSA



Fix Available  
Dec 22, 2017

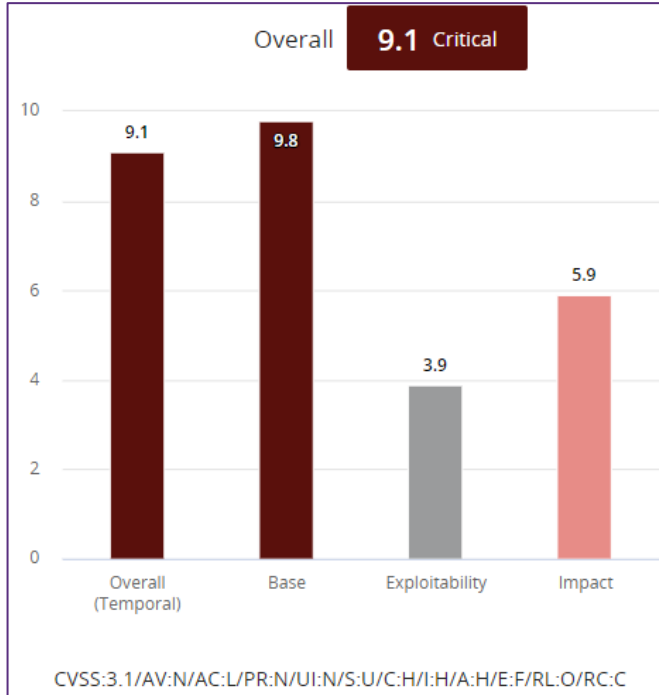


Exploit Available  
Jan 8, 2018



1,871 Days  
Vulnerability Age

Deserialization of untrusted user data in Jackson Databind could allow an attacker to perform Remote Code Execution via specially crafted JSON input. This issue exists because of an incomplete fix for CVE-2017-7525 which the vendor tried to address through an incomplete blacklist.



## Zero-click Remote Code Execution

This vulnerability can result in the execution of code on the system, triggered by a remote attacker without requiring or relying on any third party action.



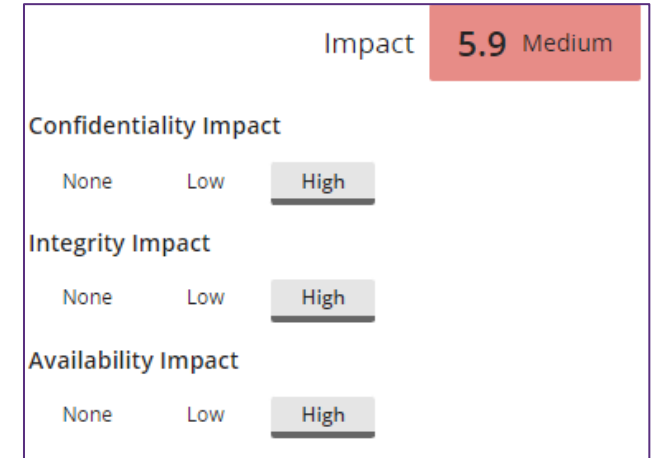
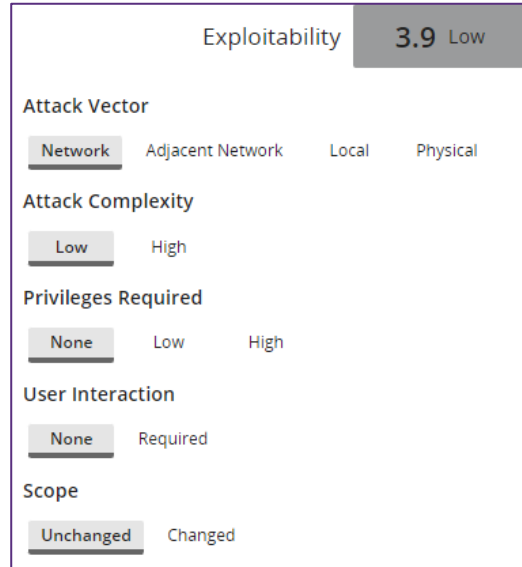
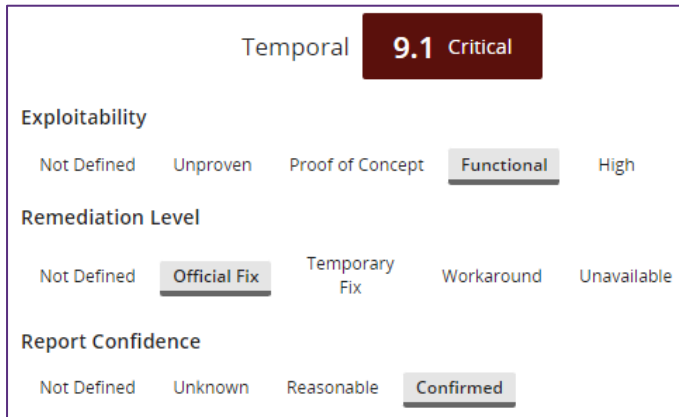
## How to fix it

### Solution - Fix Available

Fixed in 2.7.9.2, 2.8.11 and 2.9.4 by this commit.

Fixed in 2.6.7.3 by this commit.

Although this resolves the issue relating to Spring libraries present on the classpath, another attack vector exists (involving C3PO libraries) which is described in BDSA-2018-0788.





## Technical Description

An attacker can exploit this flaw by sending crafted JSON to the `readValue` method of the `ObjectMapper` class.

By taking advantage of Spring libraries available on the classpath, an attacker can construct a POP gadget chain which when des

### References and Related Links

#### Advisories

- <http://seclists.org/bugtraq/2018/Jan/28>
- <https://github.com/FasterXML/jackson-databind/issues/1855>
- <https://medium.com/@cowtowncoder/on-jackson-cves-dont-panic-here-is-what-you-need-to-know-5>

#### Vendor Upgrade

- <https://github.com/FasterXML/jackson-databind/releases/tag/jackson-databind-2.6.7.3>
- <https://github.com/FasterXML/jackson-databind/releases/tag/jackson-databind-2.7.9.2>
- <https://github.com/FasterXML/jackson-databind/releases/tag/jackson-databind-2.8.11>
- <https://github.com/FasterXML/jackson-databind/releases/tag/jackson-databind-2.9.4>

#### Patch

- <https://github.com/FasterXML/jackson-databind/commit/a3939d36edcc755c8af55bdc1969e0fa8438f>
- <https://github.com/FasterXML/jackson-databind/commit/bb45fb16709018842f858f1a6e1118676aaa3>

#### Exploit

- <https://github.com/irsl/jackson-rce-via-spel>

#### Key Events

Discovered	Discovery date not available
Vendor Notified	-
Vendor Fix	Dec 22, 2017
Disclosure	Jan 8, 2018
Vulnerability Age	1,871 Days
Exploit Available	Jan 8, 2018



# Vulnerabilities Found at CyRC

CVE	BDSA	Product	Researcher	Tool	References
<a href="#">CVE-2021-43175</a>	BDSA-2021-3657	<a href="#">GOautodial goAPI</a>	Scott Tolley	<a href="#">Seeker</a>	<a href="#">Synopsys advisory</a>
<a href="#">CVE-2021-43176</a>	BDSA-2021-3656	<a href="#">GOautodial goAPI</a>	Scott Tolley	<a href="#">Seeker</a>	<a href="#">Synopsys advisory</a>
<a href="#">CVE-2021-33177</a>	BDSA-2021-2845	<a href="#">Nagios XI</a>	Scott Tolley	<a href="#">Seeker</a>	<a href="#">Synopsys advisory</a>
<a href="#">CVE-2021-33179</a>	BDSA-2021-2847	<a href="#">Nagios XI</a>	Scott Tolley	<a href="#">Seeker</a>	<a href="#">Synopsys advisory</a>
<a href="#">CVE-2021-33178</a>	BDSA-2021-2846	<a href="#">Nagios XI</a>	Scott Tolley	<a href="#">Seeker</a>	<a href="#">Synopsys advisory</a>
<a href="#">CVE-2021-22116</a>	BDSA-2021-1329	<a href="#">RabbitMQ</a>	Jonathan Knudsen	<a href="#">Defensics</a>	<a href="#">Synopsys advisory</a>
<a href="#">CVE-2021-33175</a>	BDSA-2021-1608	<a href="#">EMQ X</a>	Jonathan Knudsen	<a href="#">Defensics</a>	<a href="#">Synopsys advisory</a>
<a href="#">CVE-2021-33176</a>	BDSA-2021-1609	<a href="#">VerneMQ</a>	Jonathan Knudsen	<a href="#">Defensics</a>	<a href="#">Synopsys advisory</a>
<a href="#">CVE-2021-3430</a>	BDSA-2021-1716	<a href="#">Zephyr Project</a>	Matias Karhumaa	<a href="#">Defensics</a>	<a href="#">Synopsys advisory</a>
<a href="#">CVE-2021-3431</a>	BDSA-2021-1718	<a href="#">Zephyr Project</a>	Matias Karhumaa	<a href="#">Defensics</a>	<a href="#">Synopsys advisory</a>
<a href="#">CVE-2021-3432</a>	BDSA-2021-1727	<a href="#">Zephyr Project</a>	Matias Karhumaa	<a href="#">Defensics</a>	<a href="#">Synopsys advisory</a>
<a href="#">CVE-2021-3433</a>	BDSA-2021-1734	<a href="#">Zephyr Project</a>	Matias	<a href="#">Defensics</a>	<a href="#">Synopsys advisory</a>

# What about SAST?

`UNSAFE_DESERIALIZATION` finds unsafe deserialization injection vulnerabilities, which arise when uncontrolled dynamic data is used within an API that can deserialize or unmarshal an object. This security vulnerability might allow an attacker to bypass security checks or to execute arbitrary code.

SAST can detect unsafe and untrusted deserialization, but the open-source component source code must be included in the analysis, not just the classes

Coverity included `jackson-databind-2.4.3.jar` (only class files) which were used for the analysis of the source captured, but not part of the captured code which contained this issue

The `DISTRUSTED_DATA_DESERIALIZATION` checker reports an issue any time distrusted data is passed into a deserialization API. An attacker who can control the deserialized object might be able to subvert aspects of the application functionality. This audit mode checker flags these code patterns for review.

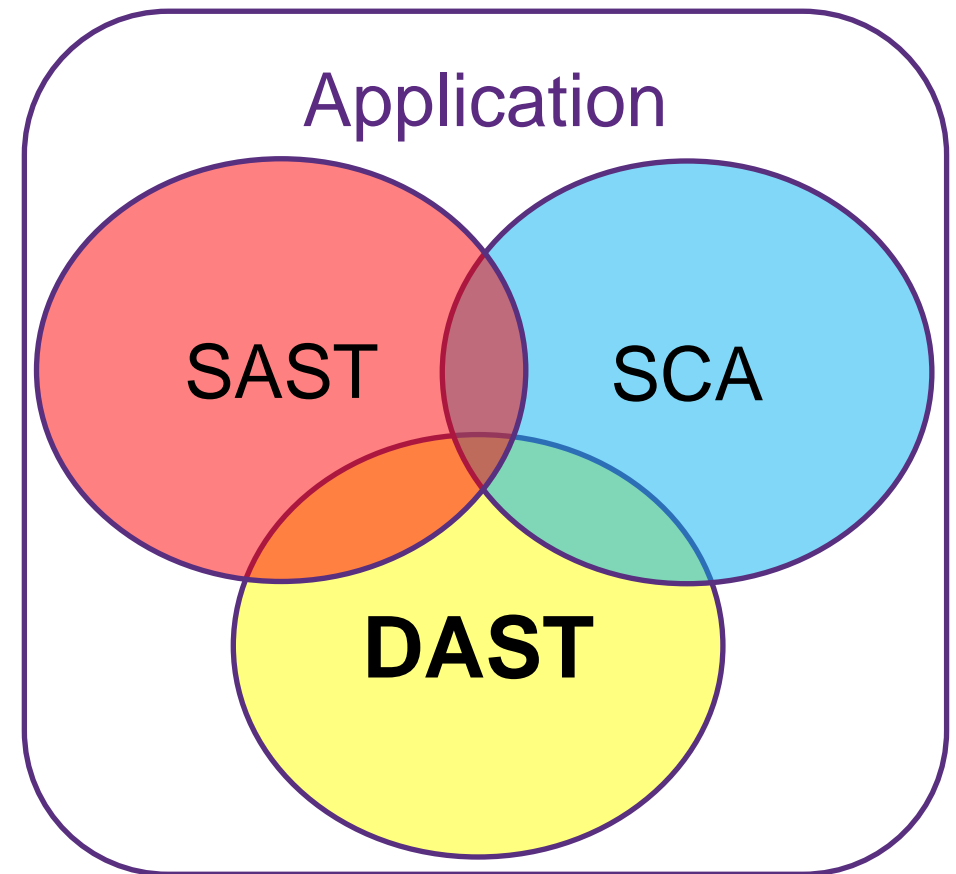
# Software Composition Analysis

## Summary

1. Use the type of scanning necessary to produce the results needed (detector, signature, binary, snippet, etc.) - some require manual validation
2. Use policies to prioritize, someone has already done the ethical hacking and risk assessment for you!
3. When a quick component update is not practical, use the research provided to understand the risk and mitigation

# Today's Agenda

1. Software Development Process
2. SAST
3. SCA
4. DAST (IAST/Fuzz)
5. ASOC
6. Help!



# Ethical Hacking (DAST)

Many tools may be used to automate ethical hacking, sending compromising data to an application and reviewing responses

Products like IAST deploy agents to watch and validate issues as web applications are exercised

Project: WebApp				Version: All	Severity: All	Status: All	Tag: All	Contains text	Q	more filters	clear	
<input type="checkbox"/> Vulnerability												
										Severity	#	Last Detected
<input type="checkbox"/>	<b>Missing Content-Type Header</b>	[Key: webapp-7]	<input checked="" type="checkbox"/> Seeker-Verified	Endpoint: /spring-angularjs-java-webapp-template-pr...	Parameter: None	Code location: None				Low	4	25 minutes ago
<input type="checkbox"/>	<b>Insecure Spring MVC Auto-Binding</b>	[Key: webapp-15]	<input checked="" type="checkbox"/> Seeker-Active-Inspection <input checked="" type="checkbox"/> Seeker-Verified	Endpoint: /spring-angularjs-java-webapp-template-pr...	Parameter: None	Code location: j.s.h.HttpServlet.service():732				Low	2	29 minutes ago
<input type="checkbox"/>	<b>Missing Content-Type-Options Header</b>	[Key: webapp-2]	<input checked="" type="checkbox"/> Seeker-Verified	Endpoint: /spring-angularjs-java-webapp-template-pr...	Parameter: None	Code location: None				Info	3	29 minutes ago
<input type="checkbox"/>	<b>Insufficient SSL Enforcement</b>	[Key: webapp-3]	<input checked="" type="checkbox"/> Seeker-Verified	Endpoint: /spring-angularjs-java-webapp-template-pr...	Parameter: None	Code location: None				High	3	29 minutes ago
<input type="checkbox"/>	<b>Missing Referrer Policy Header</b>	[Key: webapp-4]	<input checked="" type="checkbox"/> Seeker-Verified	Endpoint: /spring-angularjs-java-webapp-template-pr...	Parameter: None	Code location: None				Low	3	29 minutes ago
<input type="checkbox"/>	<b>Missing Content-Security-Policy header</b>	[Key: webapp-5]	<input checked="" type="checkbox"/> Seeker-Verified	Endpoint: /spring-angularjs-java-webapp-template-pr...	Parameter: None	Code location: None				Info	3	29 minutes ago
<input type="checkbox"/>	<b>Missing Cache Control Header</b>	[Key: webapp-6]	<input checked="" type="checkbox"/> Seeker-Verified	Endpoint: /spring-angularjs-java-webapp-template-pr...	Parameter: None	Code location: None				Info	3	29 minutes ago
<input type="checkbox"/>	<b>Missing XSS-Protection Header</b>	[Key: webapp-9]	<input checked="" type="checkbox"/> Seeker-Verified	Endpoint: /spring-angularjs-java-webapp-template-pr...	Parameter: None	Code location: None				Low	1	29 minutes ago
<input type="checkbox"/>	<b>Sensitive Data Exposed to Spellchecking Services (Based on Matchers)</b>	[Key: webapp-10]	<input checked="" type="checkbox"/> Seeker-Verified	Endpoint: /spring-angularjs-java-webapp-template-pr...	Parameter: None	Code location: o.s.w.f.OncePerRequestFilter.doFilter():114				Medium	1	29 minutes ago
<input type="checkbox"/>	<b>Sensitive Data Exposed to Spellchecking Services (Based on Input Type)</b>	[Key: webapp-11]	<input checked="" type="checkbox"/> Seeker-Verified	Endpoint: /spring-angularjs-java-webapp-template-pr...	Parameter: None	Code location: o.s.w.f.OncePerRequestFilter.doFilter():114				Low	1	29 minutes ago

The password used via the JDBC API to access the local database was weak

A weak database password allows lateral hacking from a compromised host to accessible databases

# Weak Password Used in Database Connection

WebApp / webapp-1 **Medium** ✓ Seeker-Verified

**Summary** Data Flow Verification Proof HTTP Request Previous Detections Remediation Online Training (4)



17 days ago  
LAST SEEN

Enable Project version management  
to see latest vulnerable version.  
VULNERABLE VERSION

2  
DETECTION COUNT

## Detection details [What is Weak Password Used in Database Connection?](#)

Seeker has detected that the application uses a weak password to connect to the database, which could lead to sensitive information theft and database takeover by malicious entities.

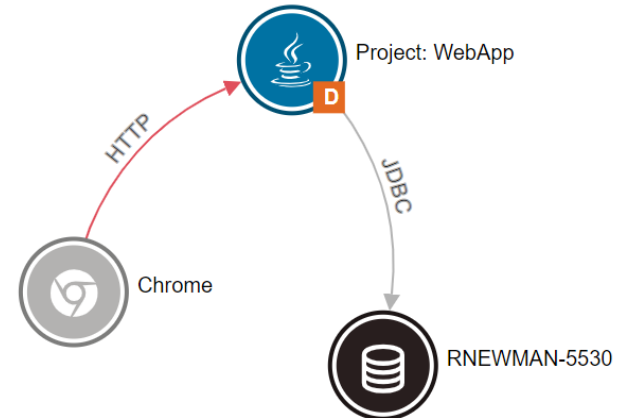
The following connection string was used by the application: jdbc:h2:mem:test

The weak password that was used with this connections string is: [masked sensitive data]

Following is the code which opened the database connection:

```
com.mchange.v2.c3p0.DriverManagerDataSource.getConnection(DriverManagerDataSource.java:134) Show stacktrace
```

```
package com.mchange.v2.c3p0;
...
class DriverManagerDataSource{
...
public void getConnection(){ ↕
...
org.h2.Driver.connect(); ↘
...
}
...
}
```





# What about SAST?

Dynamic testing *may* not have been required to detect this issue

But SAST only detected the hardcoded password in a test program (also a no-no)

The password is hardcoded in applicationContext-jdbc.xml which is ok if the file is encrypted and access is limited

The screenshot shows a SAST tool interface. At the top, there's a navigation bar with a hamburger menu, 'Issues: By Snapshot | All', and icons for help and settings. Below is a table of issues:

Component	# Items	CID	Category	Type	Standard: OWASP Web Top Ten 2021
WebApp.test	3	130125	Medium impact security	Use of hard-coded password	A2
WebApp.interceptor	6	133431	Null pointer dereferences	Dereference null return value	None
WebApp.auth	4	133520	SpotBugs: Performance	WMI: Inefficient Map Iterator	None
WebApp.Other	42				
WebApp.service	1				
WebApp.controller	13				
WebApp.unusedspringsecurity	4				
WebApp.model	1				
WebApp.core	8				

Below the table, it says '9 items match' and 'Page 1 of 1'. A dropdown shows '1 of 3 issues selected'. Below that is a code editor for 'BCryptPasswordEncoderTester.java'.

```
1 package yourwebproject2;
2
3 import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
4 import org.springframework.util.Assert;
5
6 /**
7  * @author: kameshr
8  */
9 public class BCryptPasswordEncoderTester {
10     private static BCryptPasswordEncoder passwordEncoder = new BCryptPasswordEncoder();
11
12     public static void main(String[] args) throws InterruptedException {
13         1. hardcoded_credential: The constant string "Test1234" is a credential or key.
14
15         ◆ CID 130125 (#1 of 1): Use of hard-coded password (HARDCODED_CREDENTIALS)
16         2. password_use: Calling encode. This call uses the constant string as a password. (The virtual call resolves to
17         org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder.encode(java.lang.CharSequence).)
18
19         ⚠ Passwords should be stored in a Properties file in a location that is inaccessible to unauthorized users.
20
21         String encoded = passwordEncoder.encode("Test1234");
22         Thread.sleep(5L * 1000L);
23         Assert.isTrue(passwordEncoder.matches("Test1234", encoded));
24         Assert.isTrue(!passwordEncoder.matches("Test123ss", encoded));
25         Assert.isTrue(passwordEncoder.matches("Test1234", encoded));
26
27         System.out.println("Test 1: "+passwordEncoder.matches("Test1234", encoded));
28         System.out.println("Test 2: "+passwordEncoder.matches("Test123ss", encoded));
29         System.out.println("Test 3: "+passwordEncoder.matches("Test1234", encoded));
30     }
31 }
```

# Fuzzing

Applying additional automation, a fuzzer may send all kinds of data to the application endpoints

In combination with DAST, this is a powerful detection system

```
13:34:37 Running the failing SQL injection.
13:34:37 tcp 9241 --> localhost:8009 505 request-POST-127-0-0-1-1 ANOMALY!
13:34:37 tcp 9241 <-- localhost:8009 144 response-127-0-0-1-200_Ok-1
13:34:37 opening TCP connection to localhost:8009: TCP localhost:8009
13:34:37 Running the valid case.
13:34:37 tcp 9242 --> localhost:8009 489 request-POST-127-0-0-1-1 ANOMALY!
13:34:37 tcp 9242 <-- localhost:8009 1145 response-127-0-0-1-200_Ok-1
13:34:37 opening TCP connection to localhost:8009: TCP localhost:8009
13:34:37 Running the valid SQL injection.
13:34:37 tcp 9243 --> localhost:8009 492 request-POST-127-0-0-1-1 ANOMALY!
13:34:37 tcp 9243 <-- localhost:8009 1151 response-127-0-0-1-200_Ok-1
13:34:37 Valid case(3982) size: 973 bytes
13:34:37 Passing SQL case(2245): 979 bytes
13:34:37 Failing SQL case(2245): 0 bytes
13:34:37 opening TCP connection to localhost:8009: TCP localhost:8009
13:34:37 Test case #2245 pass
13:34:37 Test case #2245 completed
13:34:51 *** Omitted logging for 1245 test cases ***
13:34:51 TEST CASE #2417
13:34:51 sample.HTTP-Request-message.request-POST-127-0-0-1-1.HTTP-Request.HTTP-Request-content.RequestLine.Request-URI.abs_pat...
13:34:51 tcp 10536 --> localhost:8009 514 request-POST-127-0-0-1-1 ANOMALY!
13:34:51 tcp 10536 <-- localhost:8009 11398 response-127-0-0-1-200_Ok-1
13:34:51 SUT responded with non HTTP message.
13:34:51 opening TCP connection to localhost:8009: TCP localhost:8009
13:34:51 Test case #2417 pass
13:34:51 Test case #2417 completed
13:35:10 *** Omitted logging for 1613 test cases ***
13:35:10 TEST CASE #567
13:35:10 sample.HTTP-Request-message.request-POST-127-0-0-1-1.HTTP-Request.HTTP-Request-content.RequestLine.element: Repeat of ...
13:35:10 tcp 12203 --> localhost:8009 1612 request-POST-127-0-0-1-1 ANOMALY!
13:35:10 tcp 12203 <-- localhost:8009 973 response-127-0-0-1-200_Ok-1
13:35:10 SUT responded with non HTTP message.
13:35:10 opening TCP connection to localhost:8009: TCP localhost:8009
13:35:10 Test case #567 pass
13:35:10 Test case #567 completed
```

## request-POST-127-0-0-1-1 [with anomaly]


```
000000 POST /\r\n
000008 mkdir:createdbyinjection-2417/Auth:HTTP/1.1\r\n
000035 Accept-Encoding: gzip,deflate\r\n
000054 Accept-Language: en-US,en;q=0.8\r\n
000075 Connection: keep-alive\r\n
00008d Content-Length: 22\r\n
0000a1 Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3\r\n
0000d1 Host: 127.0.0.1\r\n
0000e2 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/535.1 (KHTML, like Gecko) Chrome/14.0.835.163 Safari/535.1\r\n
000156 Content-Type: application/x-www-form-urlencoded\r\n
000187 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
0001d0 Cache-Control: max-age=0\r\n
0001ea \r\n
0001ec LoginPassword=Password
```

```
ies ***
-POST-127-0-0-1-1.HTTP-Request.HTTP-Request-content.RequestLine.Request-URI.abs_pat...
quest-POST-127-0-0-1-1 ANOMALY!
rponse-127-0-0-1-200_Ok-1
:8009: TCP localhost:8009
ies ***
-POST-127-0-0-1-1.HTTP-Request.HTTP-Request-content.RequestLine.element: Underflow ...
quest-POST-127-0-0-1-1 ANOMALY!
rponse-127-0-0-1-200_Ok-1
:8009: TCP localhost:8009
ies ***
-POST-127-0-0-1-1.HTTP-Request.HTTP-Request-content.RequestLine.Method: Alternative...
```

```
13:35:37 tcp 14691 --> localhost:8009 492 request-POST-127-0-0-1-1 ANOMALY!
```



# 15 new issues detected!



SEEKER  
VERIFIED

33 minutes ago

LAST  
SEEN

Enable [Project version management](#)  
to see latest vulnerable version.

VULNERABLE VERSION

1

DETECTION  
COUNT

4 minutes ago

Detection details [What is Clickjacking?](#)

Clickjacking, also known as "User Interface Redressing" is a security vulnerability that exposes the business to risks of having application users unknowingly

Summary [Data Flow](#) [Verification Proof](#) [HTTP Request](#) [Previous Detections](#) [Remediation](#) [Online Training \(2\)](#)

In order to block attempts for this attack, the web site should deny unpermitted entities to host the web site inside a frame.

It is recommended that the web site will embed the headers 'X-Frame-Options' and 'Content-Security-Policy' in **every** HTTP response sent back from the server. These headers instruct browsers to enforce improper usage of HTML frame elements that might lead to this attack.

For instance, the following headers will instruct browsers to deny hosting of web site pages inside HTML frame elements:

```
X-Frame-Options: DENY
Content-Security-Policy: frame-ancestors 'none'
```

If it is needed to host pages of the web sites inside HTML frame elements, it is recommended to use the following header values to allow such hosting only for the same origin:

```
X-Frame-Options: SAMEORIGIN
Content-Security-Policy: frame-ancestors 'self'
```

If for some reason is not possible to add the above headers, an alternative solution could be a javascript blockage of frame hosting. Following is a sample javascript code portion that blocks hosting of web pages inside frames:

```
if (top.location != self.location) {
  top.location = self.location;
}
```

Note: In some legacy browsers such as IE 7 or Safari 4.0.4, the above solution might be needed.

Code location: <i>None</i>	<a href="#">Info</a>	2	8 minutes ago
Code location: <i>None</i>	<a href="#">Info</a>	2	8 minutes ago
Code location: <i>None</i>	<span style="background-color: orange; color: white; padding: 2px 5px; border-radius: 3px;">Low</span>	2	8 minutes ago
Code location: <i>None</i>	<span style="background-color: orange; color: white; padding: 2px 5px; border-radius: 3px;">Low</span>	1	8 minutes ago
Code location: <i>None</i>	<span style="background-color: orange; color: white; padding: 2px 5px; border-radius: 3px;">Low</span>	1	8 minutes ago

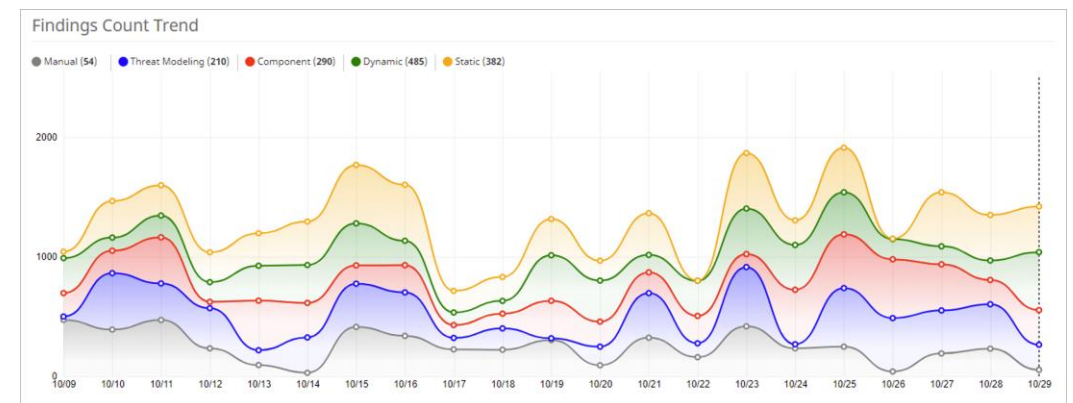
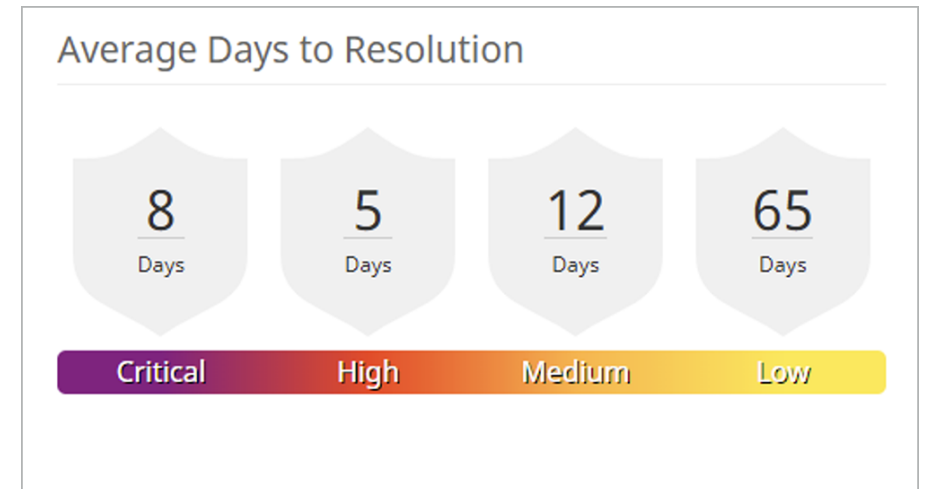
# Dynamic Application Security Testing

## Summary

- 1 Automate whenever possible
- 2 Results may take investigation
- 3 Correlation with other tools is beneficial

# Today's Agenda

1. Software Development Process
2. SAST
3. SCA
4. DAST (IAST/Fuzz)
5. ASOC
6. Help!



ID	Type	Tool	CWE	Location
17255	Using Components with Known Vulnerabilities	68 active results from Black Duck Hub	1352	Components/jackson-databind:2.4.3
17295	Credentials Management	Seeker / Weak Password Used in Database Connection	255	-
17234	Data Structure	Coverity / SIGMA.cbc_insecure_padding	461	UserController.java:141
17226	Credentials Management	Coverity / Medium impact security / Hardcoded Credentials	255	UserController.java:143
17208	Credentials Management	Coverity / Medium impact security / Hardcoded Credentials	255	BCryptPasswordEncoderTester.java:13
17288	Clickjacking	Seeker / Clickjacking	1021	/spring-angularjs-java-webapp-template-project-1.0-SNAPSHOT
17276	Clickjacking	Seeker / Clickjacking	1021	-
17240	Logging	Coverity / Audit impact security / Log injection	1210	Category
17231	Logging	Coverity / Audit impact security / Log injection	1210	UserCont
17230	Logging	Coverity / Audit impact security / Log injection	1210	UserCont
17228	Logging	Coverity / Audit impact security / Log injection	1210	UserCont
17227	Logging	Coverity / Audit impact security / Log injection	1210	UserCont
17216	Logging	Coverity / Audit impact security / Log injection	1210	NewJobS
17215	Logging	Coverity / Audit impact security / Log injection	1210	NewJobS
17212	Logging	Coverity / Audit impact security / Log injection	1210	WebAppl
17211	Logging	Coverity / Audit impact security / Log injection	1210	WebAppl
17206	Logging	Coverity / Audit impact security / Log injection	1210	Category
17204	Logging	Coverity / Audit impact security / Log injection	1210	JWTToke
17203	Logging	Coverity / Audit impact security / Log injection	1210	JWTToke
17186	Logging	Coverity / Audit impact security / Log injection	1210	WebAppl
17185	Logging	Coverity / Audit impact security / Log injection	1210	RetryJob

Lateral Hacking

Zero-Click RCE

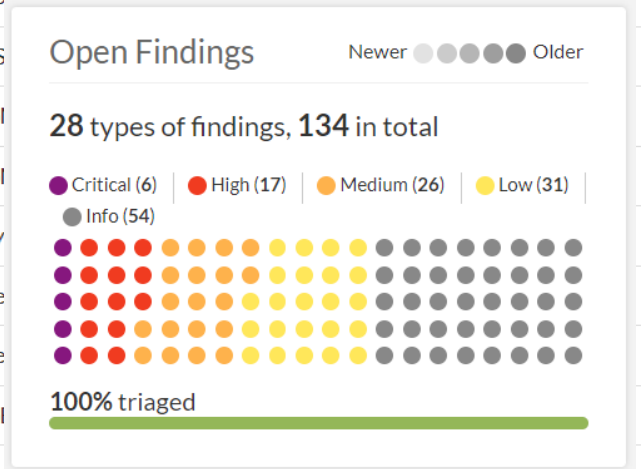
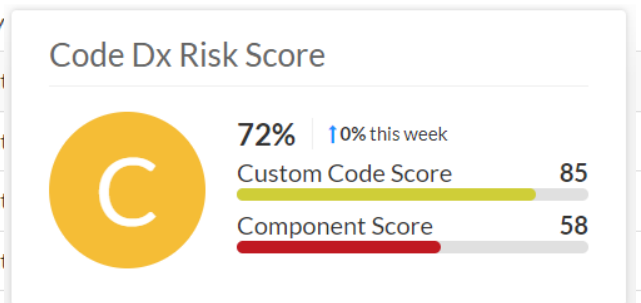
Oracle Attack

Technical Debt

Clickjacking

Logging

Issue Density Design Concern



# Today's Agenda

1. Software Development Process
2. SAST
3. SCA
4. DAST (IAST/Fuzz)
5. ASOC
6. Help!





# Thank You!

Rich Newman  
rnewman@synopsys.com  
949.466.5283

